

The Complexity of Alternating Move Games

Yaron Velner

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. We consider infinite duration alternating move games. These games were previously studied by Roth, Balcan, Kalai and Mansour [10]. They presented an FPTAS for computing an approximated equilibrium, and conjectured that there is a polynomial algorithm for finding an exact equilibrium [9]. We extend their study in two directions: (1) We show that finding an exact equilibrium, even for two-player zero-sum games, is polynomial time equivalent to finding a winning strategy for a (two-player) mean-payoff game on graphs. The existence of a polynomial algorithm for the latter is a long standing open question in computer science. Our hardness result for two-player games suggests that two-player alternating move games are harder to solve than two-player simultaneous move games, while the work of Roth et al., suggests that for $k \geq 3$, k -player games are easier to analyze in the alternating move setting. (2) We show that optimal equilibria (with respect to the social welfare metric) can be obtained by pure strategies, and we present an FPTAS for computing a pure approximated equilibrium that is δ -optimal with respect to the social welfare metric. This result extends the previous work by presenting an FPTAS that finds a much more desirable approximated equilibrium. We also show that if there is a polynomial algorithm for mean-payoff games on graphs, then there is a polynomial algorithm that computes an optimal exact equilibrium, and hence, (two-player) mean-payoff games on graphs are inter-reducible with k -player alternating move games, for any $k \geq 2$.

1 Introduction

In this work, we investigate infinitely repeated games in which players *alternate* making moves. This framework can model, for example, five telecommunication providers competing for customers: each company can observe the price that is set by the others, and it can update the price at any time. In the short term, each company can benefit from undercutting its opponents price, but since the game is repeated indefinitely, in some settings, it might be better to coordinate prices with the other companies. Such examples motivate us to study equilibria in alternating move games.

In this work, we study infinitely repeated k -player n -action games. In such games, in every *round*, a player chooses an action, and the utility of each player (for the current round) is determined according to the k -tuple of actions of the players. Each player goal is to maximize his own *long-run average* utility as the number of rounds tends to infinity.

These games were studied by Roth et al. in [10], and they showed an FPTAS for computing an ϵ -equilibrium. Their result provided a theoretical separation between the alternating move model and the simultaneous move model, since for the latter, it is known that an FPTAS for computing approximate equilibria does not exist for games with $k \geq 3$ players unless $P=PPAD$. Their result was obtained by a simple reduction to mean-payoff games on graphs. These games were presented in [5], and they play an important role in automata theory and in economics. The computational complexity of finding an exact equilibrium for such games is a long standing open problem, and despite many efforts [1–3, 6, 7, 12], there is no known polynomial solution for this problem.

We extend the work in [10] by investigating the complexity of an exact equilibrium (which was stated as an open question in [10]), and by investigating the computational complexity of finding an δ -optimal approximated equilibrium with respect to the *social welfare* metric. Our main technical results are as follows:

- We show a reduction from mean-payoff games on graphs to two-player zero-sum alternating move games, and thus we prove that k -player alternating move games are computationally equivalent to mean-payoff games on graphs for any $k \geq 2$.
- We show that optimal equilibrium can be obtained by pure strategies, and we show an FPTAS for computing an δ -optimal ϵ -equilibrium. In addition, we show that computing an exact optimal equilibrium is polynomial time equivalent to solving mean-payoff games on graphs.

We note that the first result may suggest that two-player alternating move games are harder¹ than two-player simultaneous move games, since a polynomial time algorithm to solve the latter is known [8]. Hence, along with the result of [10], we get that simultaneous move games are easier to solve with comparison to alternating move games for two-player games, and are harder to solve for $k \geq 3$ player games.

This paper is organized as following. In the next section we bring formal definitions for alternating move games and mean-payoff games on graphs. In Section 3, we show that alternating move games are at least as hard as mean-payoff games on graphs. In Section 4 we investigate the properties of optimal equilibria, and we present an FPTAS for computing an δ -optimal ϵ -equilibrium.

2 Definitions

In this section we bring the formal definitions for alternating move repeated games and mean-payoff games on graphs. Alternating move games are presented in Subsection 2.1, and mean-payoff games are presented in Subsection 2.2.

¹ Although, to the best of our knowledge, there is no hardness conjecture for mean-payoff games.

2.1 Alternating move repeated games

Actions, plays and utility function. A k -player n -action game is defined by an action set A_i for every player i , and by k utility functions, one for each player, $u_i : A_1 \times \dots \times A_k \rightarrow [-1, 1]$. W.l.o.g we assume that the size of all action set is the same, and we denote it by n . We note that any game can be rescaled so its utilities are bounded in $[-1, 1]$, however, the FPTAS that was presented in [10], and our results in Subsection 4.4 crucially rely on the assumption that the utilities are in the interval $[-1, 1]$.

An alternating move game is played for infinitely many rounds. In round t player $j = 1 + (t \bmod k)$ plays action a_j^t , and a *vector of actions* $a^t = (a_1, \dots, a_k)$ is produced, where $a_i \in A_i$ is the last action of player i . In every round t , player i receives a utility $u_i(a^t)$, which depends only in the last action of each of the k players (W.l.o.g the utility in the first k rounds is zero for all players). A sequence of infinite rounds forms a *play*, and we characterize a play either by an infinite sequence of actions or by the corresponding sequence of vectors of actions. The utility of player i in a play $a^1 a^2 \dots a^t a^{t+1} \dots$ is the *limit average payoff*, namely, $\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n u_i(a^t)$. When this limit does not exist, we define the utility of the play for player i to be $\liminf_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n u_i(a^t)$. We note that in the frame work of [10], the utility of a play was undefined when the limit does not exist. The results we present in this paper for the \liminf metric holds also for the framework of [10]. On the other hand, if we would take the \limsup value instead, then the problem is much easier, and we can easily produce a polynomial algorithm to solve these games.

Strategies. A *strategy* is a recipe for player's next action, based on the entire *history* of previous actions. Formally, a (mixed) strategy for player i is a function $\sigma_i : (A_1 \times \dots \times A_k)^* \times A_1 \times A_{i-1} \rightarrow \Delta(A_i)$, where $\Delta(S)$ denotes the set probability distribution over any finite set S . We say that σ_i is a *pure strategy* if Δ is a degenerated distribution. A *strategy profile* is a vector $\sigma = (\sigma_1, \dots, \sigma_k)$ that defines a strategy for every player. A profile of pure strategies uniquely determines the action vector in every round and yields a utility vector for the players. A profile of mixed strategies determines, for every round t in the play, a distribution of sequences of action vectors, and the *average payoff* in round t is the expected average payoff over the distribution of action vectors. Formally, for a strategy profile σ we denote the average payoff of player i in round t by

$$P_{i,t}(\sigma) = E_{a^1 a^2 \dots a^t \sim \sigma} \left[\frac{u_i(a^1) + \dots + u_i(a^t)}{t} \right]$$

and the utility of player i is $\liminf_{t \rightarrow \infty} P_{i,t}$.

Equilibria, ϵ equilibria and optimal equilibria A strategy profile forms an *equilibrium* if none of the players can strictly improve his utility (that is induced by the profile) by unilaterally deviates from his strategy (that is defined by the profile). For every $\epsilon > 0$, we say that a strategy profile forms an *ϵ -equilibrium* if none of the players can improve his utility by more than ϵ by unilaterally deviates from his strategy.

The *social welfare* of a strategy profile is the sum of the utilities of all players. An equilibrium (resp. an ϵ -equilibrium) is called an *optimal equilibrium* (optimal ϵ -equilibrium) if its social welfare is not smaller than the social welfare of any other equilibrium (ϵ -equilibrium). For $\delta > 0$, an equilibrium (resp. an ϵ -equilibrium) is called an δ -*optimal equilibrium* if its social welfare is not smaller by more than δ with comparison to the social welfare of any other equilibrium (ϵ -equilibrium).

2.2 Mean-payoff games on graphs

Plays and payoffs. A *mean-payoff game* on a graph is defined by a weighted directed bipartite graph $G = (V = V_1 \cup V_2, E, w : E \rightarrow \mathbb{Q})$ and an initial vertex $v_0 \in V$. The game consists of two players, namely, maximizer (who owns V_1) and minimizer (who owns V_2). Initially, a pebble is placed on the initial vertex, and in every round, the player who owns the vertex in which the pebble resides, advances the pebble into an adjacent vertex. This process is repeated forever and forms a *play*. A play is characterized by a sequence of edges, and the *average payoff* of a play $\rho = e_1 \dots e_t$ up to round t is denoted by $P_t = \frac{1}{t} \sum_{i=1}^t w(e_i)$. The *value* of a play is the limit average payoff (mean-payoff), namely, $\liminf_{t \rightarrow \infty} P_t$. (We note that for games on graphs, the lim sup metric gives the same complexity results.) The objective of the maximizer is to maximize the mean-payoff of a play, and the minimizer aims to minimize the mean-payoff.

Strategies, memoryless strategies, optimal strategies and winning strategies. In this work, we consider only pure strategies for games on graphs, and it is well-known that randomization does not give better strategies for mean-payoff games. A *strategy* for maximizer is a function $\sigma : (V_1 \times V_2)^* \times V_1 \rightarrow E$ that decides the next move, and similarly, for the minimizer a strategy is a function $\tau : (V_1 \times V_2)^* \rightarrow E$. A strategy is called *memoryless* if it depends only on the current position of the pebble. Formally, a memoryless strategy for the maximizer is a function $\sigma : V_1 \rightarrow E$ and similarly a memoryless strategy for the minimizer is a function $\tau : V_2 \rightarrow E$.

A profile of strategies (σ, τ) uniquely determines the mean-payoff value of a game. We say that a play $\pi = e_1 e_2 \dots e_n \dots$ is *consistent* with a maximizer strategy σ if there exists a minimizer strategy τ such that π is formed by (σ, τ) . We say that the *value* of a maximizer strategy is p if it can assure a value of at least p against any minimizer strategy. Analogously, we say that the value of a minimizer strategy is p if it can assure a value of at most p against any maximizer strategy.

We say that a maximizer strategy is *optimal* if its value is maximal (with respect to all possible maximizer strategies). Analogously, a minimizer strategy is optimal if its value is minimal. For a given threshold, we say that a maximizer strategy is a *winning strategy* if it assures mean-payoff value that is greater or equal to the given threshold, and a minimizer strategy is winning if it assures value that is strictly smaller than the given threshold.

One-player games, and games according to memoryless strategies A special (and easier) case of games on graphs is when the out-degree is one for all

the vertices that are owned by a certain player. In this case, all the *choices* are done by one player. For a two-player game on graph G , and a player-1 strategy σ , we define the one-player game graph G^σ to be the game graph that is formed by removing, for every player-1 vertex v , the out-edges that are not equal to $\sigma(v)$.

Classical results on mean-payoff games. Mean-payoff games were introduced in '79 by Ehrenfeucht and Mycielski [5], and their main result was that optimal strategies (for both players) exist, and moreover, the optimal value can be obtained by a memoryless strategy. The decision problem for mean-payoff games is to determine if the maximizer has a winning strategy with respect to a given threshold. The existence of optimal memoryless strategies almost immediately proves that the decision problem for mean-payoff games is in $\text{NP} \cap \text{coNP}$, and thus it is unlikely to be NP-hard (or coNP-hard). Zwick and Paterson [12] introduced the first pseudo-polynomial algorithm, which runs in polynomial time when the weights of the edges are encoded in unary. They also provided a polynomial algorithm for the special case of one-player mean-payoff games. A randomized sub-exponential algorithm for mean-payoff games is also known [2], but despite many efforts, the existence of a polynomial algorithm to solve mean-payoff games remains an open question, and it is one of the rare problems in computer science that is known to be in $\text{NP} \cap \text{coNP}$ but no polynomial algorithm is known.

We summarize the known results on mean-payoff games in the next theorems. The first theorem states that optimal strategies exist and moreover, there exist optimal strategies that are memoryless.

Theorem 1 ([5]) *For every mean-payoff game there exists a maximizer memoryless strategy σ and a minimizer memoryless strategy τ such that σ is optimal for the maximizer and τ is optimal for the minimizer.*

The next theorem shows that there is a polynomial algorithm that computes optimal strategies if and only if there is a polynomial algorithm for the mean-payoff games decision problem.

Theorem 2 ([12]) *The following statements are equivalent:*

- *There exists a polynomial algorithm that determines if maximizer has a winning strategy with respect to a given threshold.*
- *There exists a polynomial algorithm that determines if maximizer has a winning strategy with respect to threshold zero.*
- *There exists a polynomial algorithm that computes the optimal value that the maximizer can assure.*
- *There exists a polynomial algorithm that determines if the maximizer optimal strategy assures strictly positive value.*
- *There exists a polynomial algorithm that computes a memoryless optimal strategy for maximizer.*

3 Two-Player Zero-Sum (Alternating Move) Games are Inter-Reducible with Mean-Payoff Games

In this section we prove that there is a polynomial algorithm that computes an exact equilibrium for two-player zero-sum (alternating move) games if and only if there exists a polynomial algorithm that solves mean-payoff games.

The reduction from two-player zero-sum games to mean-payoff games is trivial, for a two-player zero-sum game with actions A_1, A_2 and utility functions $u_1 : A_1 \times A_2 \rightarrow \mathbb{Q}$ and $u_2 = -u_1$, we construct a complete bipartite game graph $G = (V = A_1 \cup A_2, E = (A_1 \times A_2) \cup (A_2 \times A_1), w : E \rightarrow \mathbb{Q})$ such that the weight of the transition from $a_1 \in A_1$ to $a_2 \in A_2$ is simply $u_1(a_1, a_2)$, and the weight of the transition from a_2 to a_1 is also $u_1(a_1, a_2)$. It is a simple observation that a pair of optimal strategies (for the maximizer and minimizer) in the mean-payoff game induces an equilibrium strategy profile in the two-player zero-sum game and vice versa.

The reduction for the converse direction is more complicated. For this purpose we bring the notion of *undirected game graph*. A mean-payoff game graph is said to be *undirected* if its edge relation is symmetric, and $w(v_1, v_2) = w(v_2, v_1)$ for every edge (v_1, v_2) . (Basically, it is a game on an undirected graph.) The next simple lemma shows a reduction from mean-payoff games on a complete bipartite undirected graphs to two-player zero-sum game.

Lemma 1 *There is a polynomial reduction from mean-payoff games on complete bipartite undirected graphs to two-player zero-sum games.*

Proof. The proof is straight forward. Let V_1 and V_2 be the maximizer and minimizer (resp.) vertices in the mean-payoff game. We construct a two-player zero-sum game in the following way. The set of action of player 1 is $A_1 = V_2$ and the set of action for player 2 is $A_2 = V_1$. We denote by W the least value for which all the weights in the undirected graphs are in $[-W, +W]$, and the utility function of player 1 is $u_1(a_1, a_2) = \frac{w(a_1, a_2)}{W} = \frac{w(a_2, a_1)}{W}$, and $u_2 = -u_1$. It is trivial to observe that an equilibrium profile induces a pair of optimal strategies for the mean-payoff game, and the proof follows. \square

Due to Lemma 1, all that is left is to prove that mean-payoff games on complete bipartite undirected graphs are equivalent to mean-payoff games. A recent result by Chatterjee, Henzinger, Krinninger and Nanongkai [4] gives us the first step towards such proof.

Theorem 3 (Corollary 24 in [4]) *Solving mean-payoff games on complete bipartite (directed) graphs is as hard as solving mean-payoff games on arbitrary graphs.*

We use the above result as a black box and extend it to complete bipartite undirected graphs. We note that the main difference between directed and undirected graphs is that for undirected graphs the weight function is symmetric. In the rest of this section we will describe a process that for a given complete

bipartite directed graph, generates a suitable symmetric weight function, and the winner in the generated graph is the same as in the original graph.

We say that a directed game graph has a *normalized weight function* if it assigns a positive weight to every out-edge of maximizer vertex, and a negative weight for every out-edge of minimizer vertex. The next lemma shows that we may assume w.l.o.g that a directed game graph has a normalized weight function.

Lemma 2 *Solving mean-payoff games on (directed) bipartite graphs is polynomial time inter-reducible to solving mean-payoff games on (directed) bipartite graphs with normalized weights.*

Proof. Let G be a non-normalized graph and let us denote by W the heaviest weight (in absolute value) that is assigned by its weight function w . We construct a normalized graph G' from G by defining a weight function w' as:

$$w'(u, v) = \begin{cases} w(u, v) + (W + 1) & \text{if } u \text{ is owned by maximizer} \\ w(u, v) - (W + 1) & \text{otherwise (if } u \text{ is owned by the minimizer)} \end{cases}$$

Clearly, G' is a normalized graph, and since G' and G are bipartite, it is straight forward to observe that for any finite path in π we have that $|w(\pi) - w'(\pi)| \leq W + 1$, and thus, for every infinite path ρ , we have that that mean-payoff value of ρ according to w is identical to the mean-payoff of ρ according to w' . Therefore, a maximizer winning (resp. optimal) strategy in graph G is a winning (optimal) strategy also in G' and vice versa, and the proof of the lemma follows. \square

In the next lemma we show that mean-payoff games on direct normalized bipartite complete graphs are as hard as mean-payoff games on undirected normalized bipartite complete graphs.

Lemma 3 *The problem of determining whether maximizer has a winning strategy for a threshold 0 for a mean-payoff games on a directed normalized bipartite complete graph is as hard as the corresponding problem for mean-payoff games on an undirected normalized bipartite complete graph*

Proof. We show that for a given directed normalized bipartite complete graph $G = (V, E, w)$ we can construct (in polynomial time) an undirected normalized bipartite complete graph $G' = (V', E', w')$ such that maximizer has a winning strategy in G (for threshold 0) if and only if he has a winning strategy in G' (again for threshold 0). Informally, we build G' from G by taking the same set of vertices and by assigning a weight $w'(u, v)$ that matches either to $w(u, v)$ or to $w(v, u)$.

To formally construct w' we present the notion of *surely losing edges*. Recall that V_1 are maximizer vertices and V_2 are minimizer vertices. We say that a maximizer vertex out-edge (v_1, v_2) is *surely losing for maximizer* if $w(v_1, v_2) + w(v_2, v_1) < 0$. We claim that if maximizer has a winning strategy, then all its memoryless winning strategies do not contain surely losing edges. That is, for every memoryless winning strategy σ and a surely losing edge (v_1, v_2) we

have $\sigma(v_1) \neq (v_1, v_2)$. Indeed, towards contradiction let us assume that $\sigma(v_1) = (v_1, v_2)$, and recall that G is a complete bipartite graph, than for the minimizer strategy τ that leads from every minimizer vertex to v_1 we have that the mean-payoff value of (σ, τ) is $\frac{w(v_1, v_2) + w(v_2, v_1)}{2} < 0$, in contradiction to the assumption that σ is a winning strategy. Analogously, (v_2, v_1) is a surely losing edge for minimizer if $w(v_1, v_2) + w(v_2, v_1) \geq 0$ and by the same arguments there is no memoryless winning strategy for minimizer that contains a surely losing edge.

We are now ready to formally define $G' = (V', E', w')$. We define $V' = V$, and since G' is complete bipartite the definition of E' follows immediately. For an edge $\{v_1, v_2\} \in E'$, where $v_i \in V_i$, we define

$$w'(v_1, v_2) = \begin{cases} w(v_1, v_2) & \text{if } w(v_1, v_2) + w(v_2, v_1) \geq 0 \\ w(v_2, v_1) & \text{otherwise (if } w(v_1, v_2) + w(v_2, v_1) < 0) \end{cases}$$

We first prove that if σ is a memoryless winning strategy for maximizer in G , then it is also a winning strategy in G' . Indeed, let ρ be a path (in G and in G') that is consistent with σ . We claim that for every path π that is a finite prefix of ρ we have $w(\pi) \leq w'(\pi)$ (that is, its weight in G' is not less than its weight in G), and we prove it by a simple induction on the length of π . For $|\pi| = 0$ the claim is trivial. For $|\pi| > 0$ let (u, v) be the last edge in π . If $u \in V_1$, then we get that $\sigma(u) = (u, v)$ and therefore (u, v) is not a surely losing edge. Therefore, by definition, $w(u, v) = w'(u, v)$ and the claim follows by the induction hypothesis (since it holds for the prefix of length $|\pi| - 1$). If $u \in V_2$, then since G is normalized we have that $w(u, v) < 0$ and $w(v, u) > 0$. Therefore, by definition, $w'(u, v) \geq w(u, v)$, and by the induction hypothesis the claim follows. Since σ is a winning strategy we get that the mean-payoff value of ρ is non-negative according to w , and by the last claim we get that the mean-payoff of ρ according to w' is greater or equal to the mean-payoff value according to w . Hence σ is a winning strategy for maximizer also in G' .

By the same arguments we get that if minimizer has a memoryless winning strategy in G , then the same strategy is winning for minimizer also in G' .

Finally, due to Theorem 1, we get that maximizer has a winning strategy in G if and only if he has a memoryless winning strategy in G if and only if he has a memoryless winning strategy in G' if and only if it has a winning strategy in G' and the proof of the lemma follows. \square

To conclude, by Lemma 1 we get that a polynomial algorithm for alternating move two-player zero-sum games exists if and only if there exists a polynomial algorithm for solving mean-payoff games on a complete bipartite undirected graph, and by Lemmas 2, 1 and 3 and by Theorem 3 we get that the latter exists if and only if there exists a polynomial algorithm for solving mean-payoff games on arbitrary (directed) graphs. Hence, the main result of this section follows.

Theorem 4 *There exists a polynomial time algorithm for computing exact equilibrium for two-player zero-sum (alternating move) games if and only if there exists a polynomial time algorithm for solving mean-payoff games on graphs.*

4 Complexity of Computing Optimal Equilibrium

In this section, we investigate the complexity of computing an optimal equilibrium. Our main results are summarized in the next theorem:

- Theorem 5**
1. *Optimal equilibrium can be obtained by a profile of pure strategies.*
 2. *If mean-payoff games are in P , then there is a polynomial algorithm for computing an exact optimal equilibrium.*
 3. *If mean-payoff games are not in P , then there is no FPTAS that approximate the social welfare of the optimal equilibrium.*
 4. *There is an FPTAS to compute an ϵ -equilibrium that is δ -optimal. (Note that it does not necessarily approximate the value of the optimal social welfare.)*

We will prove Theorem 5 in the next four subsections: In Subsection 4.1 we show the naive algorithm for computing an equilibrium that is based on Folk Theorem, and we prove basic properties of equilibria in alternating move games. In Subsection 4.2 we prove Theorem 5(1). In Subsection 4.3 we investigate the complexity of computing the social welfare of the optimal equilibrium, and prove Theorem 5(2) and Theorem 5(3). Finally, in Subsection 4.4 we prove Theorem 5(4) which is the main result of this section.

In this section, we will model n -action k -player alternating move games by a multi-weighted graph, according to the following conventions: The vertices of the graph are the vertices in the set $V = (A_1 \times A_2 \times \dots \times A_k) \times \{1, \dots, k\}$, and we say that player i owns the vertex set $V_i = (A_1 \times A_2 \times \dots \times A_k) \times \{i\}$. Intuitively, a vertex is characterized by an action vector and by a player that owns it. The pair (u, v) is in the edge relation if u is owned by player i , v is owned by player $i + 1$ (where player $k + 1$ is player 1), and there is at most one difference in the action vector of u and v and it is in position i . The weight of every edge is a vector of size k that corresponds to the utility vector of the actions. Formally, if $u = (\bar{a}_1, i)$ and $v = (\bar{a}_2, i + 1)$ then $w(u, v) = (u_1(\bar{a}_2), u_2(\bar{a}_2), \dots, u_k(\bar{a}_2))$.

For an infinite path in the multi-weighted graph we define the dimension i of *mean-payoff vector* of the path to be the mean-payoff value of the path according to dimension i . It is an easy observation that every infinite path in the graph corresponds to a play and its mean-payoff vector corresponds to the utility vector of the play. We note that the size of the graph is $k^2 \cdot n^k$ which is polynomial in the size of the encoding of the utility functions (which is $k \cdot n^k$), hence this graph can be constructed in polynomial time.

4.1 Basic properties of equilibria

The Folk Theorem gives a conceptually simple (but inefficient) technique to construct an equilibrium. Intuitively, an equilibrium is obtained when each of the players play as if the goal of all the other players is to minimize its utility, and if one of the players deviates from this strategy, then all the other players switch to playing according to a strategy that will minimize the utility of the rebellious

player. Formally, let G be the corresponding k -player game graph that models the alternating move game. For every player i , we consider a zero-sum two-player mean-payoff game graph G^i in which the maximizer owns player i vertices and the minimizer owns the other vertices. Let σ_i be an arbitrary optimal strategy for the maximizer in graph G^i , let $\overline{\sigma}_i$ be an arbitrary optimal strategy for the minimizer in G^i , and let ν_i be the value that is obtained by the strategy profile $(\sigma_i, \overline{\sigma}_i)$. Then if every player i plays according to the strategy:

If player $j \neq i$ deviated from σ_j , then play according to $\overline{\sigma}_j$ forever, and otherwise play according to σ_i

an equilibrium is formed (since by definition, playing according to σ_i assures utility at least ν_i , and deviating from σ_i assures utility at most ν_i).

In the next lemma, we can extend the basic principle of Folk Theorem and get a characterization of all the equilibriums that are obtained by a profile of pure strategies.

Lemma 4 *Let (t_1, t_2, \dots, t_k) be a utility vector such that $t_i \geq \nu_i$ (for every player i), then there exists a pure equilibrium with utility exactly t_i for every player i if and only if there exists an infinite path π in the graph G with mean-payoff vector (t_1, t_2, \dots, t_k) .*

Proof. The direction from left to right is trivial, since a profile of pure strategies has a (unique) infinite path in the graph that is consistent with the strategies. To prove the converse direction we introduce the notion of a *path strategy* and the notion of *path equilibrium*. For a path π we define the *path strategy* for player i to be: at round j (which is a player i round), play according to the j -th edge in π . And we define the strategy σ_i^π to be: If player $j \neq i$ deviated from the path strategy of π , then play forever according to $\overline{\sigma}_j$, and otherwise play according to the path strategy of π . The *path equilibrium* of the infinite path π is the profile $(\sigma_1^\pi, \dots, \sigma_k^\pi)$. The profile is an equilibrium since it assures a utility $t_i \geq \nu_i$ for every player, and if player i deviates from the strategy σ_i^π he will end up with a utility ν_i . \square

4.2 Optimal equilibrium can be obtained by pure strategies

In this subsection, we extend Lemma 4 also for the case of mixed strategies, and as a consequence we get that optimal equilibrium can be obtained by pure strategies. Intuitively, we wish to show that if a profile of (mixed) strategies yields a utility vector (t_1, \dots, t_k) , then there exists an infinite path in the graph with mean-payoff vector that is greater or equal (in every dimension) to (t_1, \dots, t_k) . Then we get that if a utility vector is obtained by a profile of mixed strategies, and then by Lemma 4 it is also obtained by a profile of pure strategies.

We formally prove the above by the next two lemmas.

Lemma 5 *Let G be a multi-weighted graph that is strongly connected, and let (t_1, \dots, t_k) be a vector. Then if for every $\alpha > 0$ there exists a (finite) cyclic path with average weight at least $t_i - \alpha$ in every dimension, then there exists an infinite path with mean-payoff vector at least (t_1, \dots, t_k) .*

Proof. We assume that for every $\alpha > 0$ there exists a finite cyclic path C_α with average weight at least $t_i - \alpha$ in every dimension, and we let v_α be an arbitrary vertex in the cycle C_α . For every two vertices u and v , we denote by $\pi_{u,v}$ the shortest path from u to v (recall that G is strongly connected), and we denote by W the size of the biggest weight (in absolute value) in G . Intuitively, we obtain an infinite path with mean-payoff vector at least (t_1, \dots, t_n) by following the cycle C_α for $\alpha = 1$, and then we follow the path $\pi_{v_\alpha, v_{\frac{\alpha}{2}}}$ and we follow the cycle $C_{\frac{\alpha}{2}}$ twice, and then we follow $C_{\frac{\alpha}{3}}$ three times and so on. However, we have to make sure that the average payoff does not decrease too much when following a cycle for the first time.

Formally, for every $i \in \mathbb{N}$, we denote by L_i the length of the cycle $C_{\frac{1}{i}}$, and by $m_i = iWL_{i+1}$. We define ρ_0 to be the empty path, and for every $i > 0$ we define $\rho_i = \rho_{i-1}\pi_{v_{\frac{1}{i-1}}, v_{\frac{1}{i}}}C_{\frac{1}{i}}^{m_i}$, and we define ρ to be the infinite path that is the limit of the sequence $\rho_0, \rho_1, \dots, \rho_i, \dots$. Due to the fact that the length of $\pi_{v_{\frac{1}{i-1}}, v_{\frac{1}{i}}}$ is bounded (by the size of the graph), and since the maximal weight is at most W (and the minimal weight is at least $-W$), we get, by a simple algebra, that the mean-payoff vector of ρ is at least (t_1, \dots, t_k) . \square

Lemma 6 *Let σ be a profile of (mixed) strategies with utility vector (t_1, \dots, t_k) . Then for every $\alpha > 0$ there exists a cyclic path in the game graph with average weight at least $t_i - \alpha$ in every dimension.*

Proof. By definition of the utility function, for every $\delta > 0$, there exists a round j such that the expected average utility, in every round after j , is at least $t_i - \delta$ in every dimension. We denote by Π_j the (finite) set of paths with length j that have non-zero probability according to the strategy profile σ , and w.l.o.g we assume that all the paths have the same probability (and a path may occur more than once in Π_j). For a path $\pi \in \Pi_j$, we denote by C_π the longest cyclic path that is a sub-path of π . We note that since G is a finite graph, we get that $|C_\pi| \geq |\pi| - 2|G|$ (where $|G|$ denotes the number of vertices in G), and in every dimension: $w(C_\pi) \geq w(\pi) - 2|G|W$ and $Avg(C_\pi) \geq Avg(\pi) - \frac{2|G|W}{|\pi| - 2|G|W} = Avg(\pi) - \frac{2|G|W}{j - 2|G|W}$. We partition Π_j to $|G|$ sets (some of them may be empty) namely Π_j^v for every $v \in G$, such that for every path $\pi \in \Pi_j^v$ we have that v is in the cyclic path C_π . For a set $\Pi_j^v = \{\pi_1, \dots, \pi_m\}$, we denote $C(\Pi_j^v) = C_{\pi_1}C_{\pi_2} \dots C_{\pi_m}$ (note that $C(\Pi_j^v)$ is a path). For every two vertices $u, v \in G$, we denote by $\pi_{u,v}$ the shortest path between u and v , and we note that $|\pi_{u,v}| \leq |G|$ and $w(\pi_{u,v}) \geq -|G|W$ in every dimension. Finally, we assume that the vertex set of G is $V = \{v_1, \dots, v_m\}$ and we define the cyclic path

$$\pi = C(\Pi_j^{v_1})\pi_{v_1, v_2}C(\Pi_j^{v_2})\pi_{v_2, v_3} \dots \pi_{v_{m-1}, v_m}C(\Pi_j^{v_m})\pi_{v_m, v_1}$$

The average weight of π in every dimension is at least

$$Avg(\Pi_j) - \frac{3|G|^2W}{j}$$

and since in every dimension $Avg(\Pi_j) \geq t_i - \delta$, then for $j \geq \frac{3|G|^2 W}{\delta}$ we get that in every dimension

$$Avg(\pi) \geq t_i - 2\delta$$

and for $\delta = \frac{\alpha}{2}$ we get that π is a cyclic path with average weight at least $t_i - \alpha$ in every dimension, and the proof of the lemma follows. \square

We are now ready to prove that the utility vector of a mixed equilibrium can be obtained by a pure equilibrium.

Proposition 1 *Let σ be a profile of mixed strategies that induces a utility vector (t_1, \dots, t_k) . Then there exists a profile σ' of pure strategies that induces exactly the same utility vector. Moreover, if σ is an equilibrium, then so is σ' .*

Proof. By Lemma 6 we get that for every $\alpha > 0$ there is a cyclic path with average weight at least $t_i - \alpha$ in every dimension. Therefore, by Lemma 5 we get that there is an infinite path in G with mean-payoff vector at least (t_1, \dots, t_k) , and by Lemma 4 we get that there is a profile of pure strategies that has utility at least (t_1, \dots, t_k) . If σ is an equilibrium we get that $t_i \geq \nu_i$ (since otherwise, player i would deviate to strategy σ_i), and thus, by Lemma 4, we get that there is a pure equilibrium that gives the same utility vector. \square

The next corollary immediately follows from Proposition 1.

Corollary 1 (Theorem 5(1)) *An optimal equilibrium can be obtained by a profile of pure strategies.*

4.3 The complexity of computing the social welfare of the optimal (exact) equilibrium

In this section we show that if there is a polynomial algorithm for mean-payoff games, then there is a polynomial algorithm to compute an optimal equilibrium in a k -player alternating move games. We also prove the converse direction, that is, we show that if there is a polynomial algorithm that computes the social welfare of the optimal equilibrium, then there is a polynomial algorithm that solves mean-payoff games. We prove these two assertions in the next two lemmas.

Lemma 7 *Suppose that mean-payoff games are in P , then there is a polynomial algorithm that computes the social welfare of an optimal equilibrium.*

Proof. Due to Corollary 1, it is enough to consider only pure strategies, and due to Lemma 4 a vector of utilities (t_1, \dots, t_k) is obtained by a pure equilibrium if and only if $t_i \geq \nu_i$ (for $i = 1, \dots, k$) and there is an infinite path in the game graph with mean-payoff (t_1, \dots, t_k) . Since we assume that there is a polynomial algorithm for computing ν_i , our problem boils down to

Find the maximal value of $\sum_{i=1}^k t_i$ subject to
 – $t_i \geq \nu_i$; and

- there exists an infinite path with mean-payoff vector at least (t_1, \dots, t_k)

It was shown in [11] (in the proof of Theorem 18) that the problem of deciding whether there exists an infinite path with mean-payoff vector at least (t_1, \dots, t_k) can be reduced (in polynomial time) to a set of linear constraints. Moreover, the generated set of constraints remain linear even when t_i is a variable. Hence, we can find a feasible threshold vector (t_1, \dots, t_k) (that is, a vector that is realizable by an infinite path in the graph) that maximizes $\sum_{i=1}^k t_i$ by linear programming. Therefore, if we have a polynomial algorithm that computes ν_i , then we can find the social welfare of the optimal equilibrium in polynomial time. \square

Lemma 7 proves Theorem 5(2) and gives an upper bound to the complexity of computing optimal equilibrium. In the next lemma we show that this bound is tight, and that the social welfare of the optimal equilibrium cannot be approximated, unless mean-payoff games are in P.

Lemma 8 (Theorem 5(3)) *There is no FPTAS that approximates the social welfare of an optimal equilibrium, unless mean-payoff games are in P.*

Proof. Due to Theorem 2 and Theorem 4 it is enough to show that if we could approximate the social welfare of the optimal equilibrium in a three-player game, then we would be able to determine whether in a two-player zero-sum game, player 1 has a strategy that assures a value that is strictly greater than 0. The proof is straight forward. Let (A_1, A_2) be the actions of a zero-sum two-player game with utility functions $u_1 : A_1 \times A_2 \rightarrow [-1, 1]$ and $u_2 = -u_1$. We construct a three-player game $(A'_1, A'_2, A'_3, u'_1, u'_2, u'_3)$ in the following way:

- $A'_1 = A_1 \cup \{\$ \}$, $A'_2 = A_2 \cup \{\$ \}$ and $A'_3 = \{\$ \}$ (where $\$$ is a fresh action).
- Let a_2^* be an arbitrary action in $A'_2 - \{\$ \}$. We define the utility function u'_1 to be

$$u'_1(a_1, a_2, a_3) = \begin{cases} u_1(a_1, a_2) & \text{if } a_1 \neq \$ \text{ and } a_2 \neq \$ \\ u_1(a_1, a_2^*) & \text{if } a_1 \neq \$ \text{ and } a_2 = \$ \\ 0 & \text{otherwise (if } a_1 = \$) \end{cases}$$

We define $u'_2 = -u'_1$, and

$$u'_3(a_1, a_2, a_3) = \begin{cases} 1 & \text{if } a_1 = \$ \\ -1 & \text{otherwise (if } a_1 \neq \$) \end{cases}$$

The reader can verify that if player 1 has a strategy to assure utility greater than 0 in the zero-sum game, then in any profile of equilibrium in the three-player game he will play $\$$ only for a negligible number of rounds, and the social welfare of the equilibrium will be -1 . On the other hand, if player 1 cannot assure utility at least 0, then a profile of strategies in which all three players play $\$$ forever is an equilibrium and its social welfare is 1. Hence, even for $\epsilon = 1$ we cannot approximate the social welfare of the optimal equilibrium, unless mean-payoff games are in P. \square

4.4 An FPTAS to compute an ϵ -equilibrium that is δ -optimal

In this subsection, we assume that the utilities of the players are scaled to rationals in $[-1, 1]$, and we will describe an algorithm that computes an ϵ -equilibrium that is δ -optimal (with respect to all ϵ -equilibria) and runs in time complexity that is polynomial in the input size and in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. Subsection 4.3 suggests that in order to compute a δ -optimal ϵ -equilibrium we should approximate (by some value) the values of ν_1, \dots, ν_k and then compute the optimal infinite path (with respect to the sum of utilities) that has utility for player i that is greater than the approximation of ν_i . However, this approach would not work, since the optimal social welfare is not a continuous function with respect to the values ν_1, \dots, ν_k .

We denote by OPT_ϵ the social welfare of the optimal ϵ -equilibrium. We base our solution on the next lemma, which gives two key properties of OPT_ϵ .

Lemma 9 1. If $\epsilon_1 \geq \epsilon_2$, then $OPT_{\epsilon_1} \geq OPT_{\epsilon_2}$.
 2. For every $\alpha \in [0, 1]$ and $\epsilon_1, \epsilon_2 > 0$, let $\epsilon = \alpha\epsilon_1 + (1 - \alpha)\epsilon_2$, then there exists an ϵ -equilibrium with social welfare

$$\alpha OPT_{\epsilon_1} + (1 - \alpha) OPT_{\epsilon_2}$$

Proof. The first item of the lemma is a trivial observation. In order to prove the second item, we observe that by Lemma 4 (and since by Proposition 1 it is enough to consider only pure equilibria) it is enough to prove that there is an infinite path π with utility at least $\nu_i - \epsilon$ in every dimension and with social welfare $\alpha OPT_{\epsilon_1} + (1 - \alpha) OPT_{\epsilon_2}$. By Proposition 1, it is enough to show that there is a profile σ of mixed strategies (that need not be an equilibrium) that has a utility at least $\nu - \epsilon$ in every dimension and has a social welfare at least $\alpha OPT_{\epsilon_1} + (1 - \alpha) OPT_{\epsilon_2}$. The construction of σ is trivial. For $i = 1, 2$, let σ_{ϵ_i} be a profile of strategies that induces an ϵ_i -optimal equilibrium, then we construct σ by playing according to σ_{ϵ_1} with probability α and playing according to σ_{ϵ_2} with probability $1 - \alpha$. \square

Corollary 2 For every $\zeta \leq \frac{\epsilon\delta}{4k}$ we have

$$OPT_{\epsilon+\zeta} - \frac{\delta}{2} \leq OPT_\epsilon \leq OPT_{\epsilon+\zeta}$$

Proof. The fact that $OPT_\epsilon \leq OPT_{\epsilon+\zeta}$ follows immediately from Lemma 9(1). To prove that $OPT_{\epsilon+\zeta} - \delta \leq OPT_\epsilon$, we set $\alpha = \frac{\zeta}{\epsilon}$ and by Lemma 9(2), and since $\alpha\zeta + (1 - \alpha)(\epsilon + \zeta) = \epsilon$ we get

$$\alpha OPT_\zeta + (1 - \alpha) OPT_{\epsilon+\zeta} \leq OPT_\epsilon$$

since the utility function is scaled to $[-1, 1]$ we get that $OPT_\zeta \geq -k$ and $OPT_{\epsilon+\zeta} \leq k$, and thus we have

$$-2k\alpha + OPT_{\epsilon+\zeta} \leq OPT_\epsilon$$

since $\alpha = \frac{\zeta}{\epsilon}$ we get

$$\frac{-2k\zeta}{\epsilon} + OPT_{\epsilon+\zeta} \leq OPT_{\epsilon}$$

and since $\zeta \leq \frac{\epsilon\delta}{4k}$ we have

$$OPT_{\epsilon+\zeta} - \frac{\delta}{2} \leq OPT_{\epsilon}$$

□

By the above corollary, to approximate OPT_{ϵ} , it is enough to approximate by $\frac{\delta}{2}$ the value of $OPT_{\epsilon+\zeta}$ for some $\zeta \leq \frac{\epsilon\delta}{4k}$. For this purpose, we extend the notion of ϵ -equilibrium also for k -dimensional vectors, and we say that a profile of strategies is a $\bar{\beta}$ -equilibrium if player i cannot improve its utility by at least β_i . Let us denote by $\min \bar{\beta}$ and by $\max \bar{\beta}$ the minimal and maximal element of $\bar{\beta}$ (respectively). Then by definition,

$$OPT_{\min \bar{\beta}} \leq OPT_{\bar{\beta}} \leq OPT_{\max \bar{\beta}}$$

and by Corollary 2 we get that

$$OPT_{\bar{\beta}} \leq OPT_{\max \bar{\beta}} \leq OPT_{\bar{\beta}} + (\max \bar{\beta} - \min \bar{\beta}) \cdot \frac{4k}{\epsilon} \quad (1)$$

We are now ready to present an FPTAS that computes a δ -approximation for OPT_{ϵ} : (1) Set $\zeta = \frac{\epsilon\delta}{4k}$, and compute a ζ approximation of ν_i for every player i , and denote it by r_i . (2) Compute the optimal path π (with respect to social welfare) that has utility at least $r_i - (\epsilon - \zeta)$ for every player, and return its social welfare.

We note that we can execute the first step of the algorithm in polynomial time due to [10](Observation 3.1), and we can execute the second step in polynomial time by solving the linear programming problem that we described in the proof of Lemma 7. The next lemma proves the correctness of our approximation algorithm.

Lemma 10 *Let $S(\pi)$ denote the social welfare of π . Then $S(\pi) - \delta \leq OPT_{\epsilon} \leq S(\pi)$*

Proof. We denote player- i utility according to π by $u_i(\pi)$ and we construct the vector $\bar{\beta}$ by defining

$$\beta_i = \begin{cases} \nu_i - u_i(\pi) & \text{if } \nu_i - u_i(\pi) \geq (\epsilon - \frac{\zeta}{2}) \\ \epsilon - \frac{\zeta}{2} & \text{otherwise} \end{cases}$$

and we claim that π is an optimal $\bar{\beta}$ -equilibrium. The claim holds because for every $\bar{\beta}$ -equilibrium, the utility of player i is at least $\nu_i - (\epsilon - \zeta)$, and π is the optimal path with utility at least $\nu_i - (\epsilon - \zeta)$ for every player. In addition, we claim that $\max \bar{\beta} - \min \bar{\beta}$ is at most $\frac{\zeta}{2}$. The claim holds because by the construction

of β we have that $\min \bar{\beta} \geq \epsilon - \frac{\zeta}{2}$, and $\nu_i - u_i(\pi)$ is at most $\epsilon - \zeta$ (since π is an $(\epsilon - \zeta)$ -equilibrium). Hence $\max \bar{\beta} - \min \bar{\beta} \leq \frac{\zeta}{2}$. Therefore, since $\max \bar{\beta} \leq \epsilon$ and by Equation 1, we get that

$$S(\pi) \leq OPT_\epsilon \leq S(\pi) + \frac{\zeta}{2} \cdot \frac{4k}{\epsilon\delta}$$

and since $\zeta = \frac{\epsilon\delta}{4k}$ we get that

$$S(\pi) \leq OPT_\epsilon \leq S(\pi) + \frac{\delta}{2}$$

and the assertion of the lemma follows. \square

Lemma 10 along with the complexity analysis that we provided, proves that there is an FPTAS to compute an ϵ -equilibrium that is δ -optimal, and Theorem 5(4) follows. We also note that our proof for Theorem 5(4) gives a constructive (and polynomial) algorithm that computes a description of an actual ϵ -equilibrium that is δ -optimal.

Acknowledgement

This work was carried out in partial fulfillment of the requirements for the course Computational Game Theory that was given by Prof. Amos Fiat in Tel Aviv University.

References

1. Rick L. Andrews, Ajay K. Manrai, and Nikolai Pissaruk. Mean cost cyclical game. *Math. Oper. Res.*, 24(4):817–828, November 1999.
2. Henrik Björklund and Sergei Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210 – 229, 2007. [jce:title;29th Symposium on Mathematical Foundations of Computer Science MFCS 2004;ce:title;.](#)
3. Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.
4. Krishnendu Chatterjee, Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Polynomial-time algorithms for energy games with special weight structures. In *ESA*, pages 301–312, 2012.
5. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
6. V.A. Gurvich, A.V. Karzanov, and L.G. Khachivan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28(5):85 – 91, 1988.
7. Y.M. Lifshits and D.S. Pavlov. Potential theory for mean payoff games. *Journal of Mathematical Sciences*, 145:4967–4974, 2007.

8. Michael L. Littman and Peter Stone. A polynomial-time nash equilibrium algorithm for repeated games. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 48–54, 2004.
9. Aaron Roth. Personal website.
<http://www.cs.cmu.edu/~alroth/alternatingmoves.html>.
Accessed: 25/12/2012.
10. Aaron Roth, Maria Florina Balcan, Adam Kalai, and Yishay Mansour. On the equilibria of alternating move games. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 805–816, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.
11. Michael Ummels and Dominik Wojtczak. The complexity of nash equilibria in limit-average games. In *CONCUR*, pages 482–496, 2011.
12. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.